Optimization of Multi-Project Multi-Site Location Based on MOPSOs

ZHANG Yong, GONG Dun-wei, ZHOU Yong

School of Information and Electronic Engineering, China University of Mining & Technology, Xuzhou, Jiangsu 221008, *China*

Abstract: Multi-project multi-site location problems are multi-objective combinational optimization ones with discrete variables which are hard to solve. To do so, the case of particle swarm optimization is considered due to its useful characteristics such as easy implantation, simple parameter settings and fast convergence. First these problems are transformed into ones with continuous variables by defining an equivalent probability matrix in this paper, then multi-objective particle swarm optimization based on the minimal particle angle is used to solve them. Methods such as continuation of discrete variables, update of particles for matrix variables, normalization of particle position and evaluation of particle fitness are presented. Finally the efficiency of the proposed method is validated by comparing it with other methods on an eight-project-ten-site location problem.

Key words: multi-project location problems; multi-objective optimization; particle swarm optimization

CLC number: TP 301

1 Introduction

Multi-project multi-site location problems are wide-spread in practical engineering and city planning. A multi-objective fuzzy dynamic planning [1] and a decision-making method based on a graph of the network [2] were presented by Zhou et al to solve the location problems for a few engineering projects. These problems were also solved using the traditional Hungary decision-making method [3]. A model for a multi-objective decision-making method for site selecting of projects based on the principle of rewarding good and punishing bad was proposed by Dai et al [4]. However, an exact description of objective weights is necessary for this method. And in general, decision makers can only offer a rough description of objective weights.

Particle swarm optimization (PSO) is a population-based stochastic optimization technique proposed by Kennedy et al which imitates the social behavior of a bird flock flying around and sitting down on a pylon [5]. Because the PSO has such good performance characteristics as easy implantation, simple parameter settings and fast convergence, it has been successful in optimizing practical problems. So it has been used to solve the TSP problem with discrete variables by defining a fuzzy matrix [6]. Its application in multi-objective optimization can be seen in references[7-11].

In this paper a multi-objective particle swarm optimization based on the minimal particle angle (MOPSO)[11] is used to optimize multi-project multi-site location problems by presenting such new operators as continuation of discrete variables, normalization of particle position and evaluation of particle fitness. Finally the proposed method in this paper is compared with the method presented in reference [4] on an eight-project-ten-site location problem.

2 Multi-Project-Multi-Site Location Problems

Let $U=\{U_1,U_2,\cdots,U_m\}$ and $V=\{V_1,V_2,\cdots,V_n\}$ stand for the sets of projects and sites respectively. The optimized objective functions are $f=\{f_1,f_2,\cdots,f_l\}$,

and $A^k = \begin{array}{c|cccc} & V_1 & V_2 & \cdots & V_n \\ \hline U_1 & a_{11}^k & a_{12}^k & \cdots & a_{1n}^k \\ U_2 & a_{21}^k & a_{22}^k & \cdots & a_{2n}^k \\ \vdots & \vdots & \vdots & & \vdots \\ U_m & a_{m1}^k & a_{m2}^k & \cdots & a_{mn}^k \end{array}$ is a benefit matrix with f_k, where a_{ij}^k is a benefit value with f_k when project U_i is built on site V_j, (i=1, 2,$\cdots$, m; $j=1,2,\cdots,n$; $k=1,2,\cdots,l$; $n\ge m$ [4]).

For f_k, the purpose of location problems is to

Received 01 November 2005; accepted 14 December 2005

Project 60304016 supported by the Nationa Natural Science Foundation of China

Corresponding author. Tel: +86-516-83995199; E-mail address: yongzh401@126.com

look for the best collocation $X \in R^{m \times n}$ with $\{U_i, V_j\}$ to make $f_k(x) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} a_{ij}^k$ have the best value, where $X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}$ is a solution matrix, and $x_{ij} = \begin{cases} 1 & U_i \text{ is built on } V_j \\ 0 & U_i \text{ isn't built on } V_j \end{cases}$.

In the case of one site V_j, where no project or only one project is to be built on that site, we have $\sum_{r=1}^{m} x_{rj} \le 1$; in the case of project U_i, where one project can be built on only one site, then we have $\sum_{s=1}^{n} x_{is} = 1$.

For f, the purpose of location problems is to look for the best collocation with $\{U_i, V_j\}$ to make all elements of f have the optimal value. Taking the minimization problem for example, its mathematical model is described as follows:

$$\min f(X) = \min\{f_1(X), f_2(X), \cdots, f_l(X)\} \quad (1)$$

$$\text{s.t.} \quad \sum_{r=1}^{m} x_{rj} \le 1, \quad \sum_{s=1}^{n} x_{is} = 1$$

where $f_k(X) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} a_{ij}^k$, $k = 1, 2, \cdots, l$.

3 Optimization of Multi-project Multi-site Location Problems

In this section, multi-objective particle swarm optimization based on the minimal particle angle is used to optimize multi-project multi-site location problems. First, multi-objective combinational optimization problems with discrete variables are transformed into the ones with continuous variables by an equivalent probability matrix. Then a MOPSO method is used to optimize multi-objective multi-project location problems, where the main problems to be solved are 1) continuation of discrete variables, 2) update of particles for matrix variables, 3) normalization of the particle positions, and 4) evaluation of particle fitness.

3.1 Continuation of discrete variables

A matrix X describes the relationship between projects and sites correctly, but it isn't suitable for the MOPSO. An equivalent probability matrix is defined to encode location problems in this subsection.

Definition (equivalent probability matrix): A matrix $P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mn} \end{bmatrix}$ is called the equivalent probability matrix if it satisfies:

1) $\sum_{j=1}^{n} P_{ij} = 1$ and

2) $P_{ij} \in [0,1]$, $i = \{1, 2, \cdots, m\}$, $j = \{1, 2, \cdots, n\}$,

where P_{ij} indicates a probability that U_i is built on V_j.

3.2 Update of particles

Let P be an optimized decision variable for particle positions. Then the corresponding particle velocities are $V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix}$.

For the i-th particle at generation t, its position, globally optimal particle and locally optimal particle are matrix P_i^t, G_i^t and L_i^t respectively. Then a formula for updating particles is as follows:

$$\begin{cases} V_i^{t+1} = w \otimes V_i^t + c_1 r_1 \otimes (L_i^t - P_i^t) + c_2 r_2 \otimes (G^t - P_i^t), \\ X_i^{t+1} = P_i^t + V_i^{t+1} \end{cases} \quad (2)$$

where $i = 1, 2, \cdots, N$, N is the swarm size, c_1 and c_2 and are two positive constants, r_1 and r_2 and are two random numbers within [0,1], and w is an inertia weight.

3.3 Normalization of particle position

The position matrix P_i^t may disagree with the definition of the equivalent probability matrix after some generations via formula (2), so it is necessary to normalize the particle positions. Supposing the s-th row of P_i^t is $(P_{i,s1}^t, P_{i,s2}^t, \cdots, P_{i,sn}^t)$, then a formula of normalization is described as follows:

$$P_{i,sj}^{t\,\prime} = \frac{P_{i,sj}^t - \min\limits_{k \in \{1,2,\cdots,n\}} P_{i,sk}^t}{\sum_{r=1}^{n}(P_{i,sr}^t - \min\limits_{k \in \{1,2,\cdots,n\}} P_{i,sk}^t)} \quad (3)$$

where $\min\limits_{k \in \{1,2,\cdots,n\}} P_{i,sk}^t$ is the minimum in

$(P_{i,s1}^{t}, P_{i,s2}^{t}, \cdots, P_{i,sn}^{t}), s = 1, 2, \cdots, m.$

3.4 Decoding and evaluation of particle fitness

In this subsection a method of decoding is proposed. First, look for the maximum element of the first row and let its value be 1. Other elements of the column and row which include the above max element are 0. Then evaluate the elements of the second row and correlative column using the same method; the rest may be deduced by analogy. Finally, a solution matrix with all elements belonging to $\{0,1\}$ can be obtained.

P_i^t is transformed into a solution matrix X_i^t according to the above method, so we may evaluate particle fitness by formula (1) and update the globally optimal particle and the locally optimal particle.

3.5 Steps of algorithms

The steps of MOPSO on multi-project multi- site location problems are described as follows:

Step 1: Initialize particle positions according to the definition of equivalent probability matrixes, locally optimal particles, the archive and the max times of iterations $T_{\max}$, and particles' velocities;

Step 2: Decode and evaluate the particle fitness according to the method given in subsection 3.4;

Step 3: Update the archive, the globally optimal particles and the locally optimal particles according to the method proposed in reference [11];

Step 4: Generate offspring particles via formula (2), and normalize the particle positions according to the method given in subsection 3.3;

Step 5: Judge whether $T_{\max}$ is met or not. If yes, stop the algorithm, otherwise go to step 2.

4 Results

4.1 Description of problems and parameter settings

In this subsection an eight-project-ten-site location problem is used to validate the method presented in this paper, where the optimized objectives include maximizing economic benefit f_1 of the projects, comprehensive environment index f_2, and social benefit f_3 of the projects, and their corresponding benefit matrixes $\boldsymbol{A}_1$, $\boldsymbol{A}_2$, $\boldsymbol{A}_3$ are given as follows, respectively:

$$A_1=\begin{bmatrix} 16.63 & 9.50 & 3.33 & 3.08 & 4.63 & 19.32 & 5.21 & 3.57 & 12.45 & 1.37 \\ 18.15 & 8.54 & 10.51 & 16.70 & 11.31 & 4.62 & 3.09 & 15.90 & 4.84 & 0.46 \\ 16.55 & 16.32 & 18.48 & 8.56 & 6.90 & 11.79 & 14.51 & 4.29 & 8.23 & 9.95 \\ 0.94 & 17.80 & 11.25 & 18.16 & 10.82 & 9.44 & 6.64 & 17.45 & 1.06 & 9.84 \\ 2.19 & 10.91 & 13.14 & 18.79 & 10.07 & 2.85 & 17.33 & 9.76 & 19.61 & 18.18 \\ 16.72 & 2.67 & 7.86 & 1.04 & 18.40 & 7.70 & 16.19 & 18.00 & 1.06 & 12.47 \\ 8.00 & 13.40 & 19.41 & 10.54 & 12.50 & 13.23 & 0.81 & 1.90 & 19.29 & 16.03 \\ 9.88 & 1.63 & 10.14 & 12.58 & 15.75 & 14.61 & 18.20 & 10.16 & 8.36 & 3.68 \end{bmatrix}$$

$$A_1=\begin{bmatrix} 0.22 & 2.31 & 1.29 & 1.22 & 0.30 & 1.90 & 2.73 & 0.86 & 2.04 & 1.22 \\ 2.70 & 2.88 & 1.52 & 2.64 & 2.95 & 2.54 & 1.43 & 1.59 & 2.30 & 0.06 \\ 0.43 & 1.17 & 0.65 & 0.10 & 0.24 & 1.22 & 1.44 & 0.14 & 1.95 & 0.74 \\ 1.28 & 0.75 & 2.35 & 1.02 & 1.90 & 2.44 & 0.51 & 2.29 & 2.23 & 2.99 \\ 1.13 & 1.93 & 1.81 & 1.08 & 2.86 & 0.68 & 0.75 & 2.49 & 1.57 & 1.18 \\ 1.98 & 2.97 & 0.68 & 2.69 & 1.09 & 1.12 & 1.30 & 1.21 & 0.82 & 1.43 \\ 2.09 & 0.49 & 2.34 & 2.93 & 2.64 & 1.00 & 0.06 & 2.67 & 0.53 & 1.33 \\ 2.68 & 1.26 & 0.96 & 1.06 & 0.84 & 2.52 & 0.90 & 0.91 & 0.96 & 0.98 \end{bmatrix}$$

$$A_3=\begin{bmatrix} 9.00 & 5.61 & 3.51 & 9.72 & 6.42 & 8.14 & 7.96 & 7.95 & 3.51 & 2.95 \\ 3.76 & 6.02 & 9.79 & 4.92 & 7.52 & 1.52 & 3.29 & 6.75 & 7.80 & 6.78 \\ 2.04 & 7.51 & 7.49 & 5.25 & 2.80 & 3.19 & 5.05 & 0.49 & 7.83 & 1.56 \\ 8.07 & 0.25 & 1.37 & 9.04 & 3.65 & 6.98 & 0.60 & 0.71 & 9.45 & 4.46 \\ 0.57 & 6.69 & 2.97 & 0.01 & 2.36 & 2.00 & 2.69 & 7.42 & 6.44 & 2.02 \\ 2.58 & 0.63 & 6.70 & 3.18 & 8.25 & 9.53 & 4.53 & 0.47 & 3.91 & 0.30 \\ 6.96 & 3.37 & 7.16 & 6.57 & 6.56 & 2.47 & 7.70 & 1.47 & 1.48 & 6.20 \\ 1.42 & 6.44 & 1.77 & 0.88 & 5.68 & 0 & 6.21 & 6.02 & 5.08 & 8.26 \end{bmatrix}$$

The values of the pertinent parameters are set as Table 1.

Table 1 Parameter settings

Parameters	Values
Size of swarm N	40
$T_{\max}$	200
w	0.4
c_1, c_2	2
Size of archive	10

4.2 Experimental results and comparison

The method proposed in this paper is compared with the method presented in reference [4] on an eight-project-ten-site location problem in order to validate its efficiency. In the paper the goal is to minimize function vector $\boldsymbol{f} = (-f_1, -f_2, -f_3)$.

If the supposed objective weights are $\boldsymbol{W} = (1/3, 1/3, 1/3)$, we can get the benefit matrix $\boldsymbol{J}$ with the method proposed in reference[4].

$$J=\begin{bmatrix} 0.0984 & 0.1142 & -0.1830 & -0.0393 & -0.1944 & \mathbf{0.3060} & 0.1274 & -0.1156 & 0.0915 & -0.2414 \\ \mathbf{0.1375} & 0.0558 & 0.0856 & 0.1439 & 0.1528 & -0.1623 & -0.2298 & 0.0994 & -0.0038 & -0.2702 \\ -0.0917 & 0.2589 & 0.1815 & -0.1447 & -0.2369 & 0.0432 & 0.1945 & -0.3707 & \mathbf{0.2836} & -0.1402 \\ -0.0600 & -0.1250 & -0.0430 & \mathbf{0.2081} & -0.0108 & 0.1389 & -0.3089 & 0.0258 & 0.0714 & 0.1017 \\ -0.3051 & 0.1784 & 0.0223 & -0.1174 & 0.0556 & -0.2785 & -0.0480 & \mathbf{0.2479} & 0.2455 & -0.0254 \\ 0.0928 & -0.1041 & -0.0205 & -0.0539 & \mathbf{0.2410} & 0.1140 & 0.0917 & -0.0516 & -0.2201 & -0.1181 \\ 0.0639 & -0.1379 & \mathbf{0.2558} & 0.1679 & 0.1696 & -0.1199 & -0.2056 & -0.1383 & -0.1119 & 0.0759 \\ 0.0446 & -0.0508 & -0.1431 & -0.1236 & 0.0821 & 0.0549 & \mathbf{0.1452} & 0.0168 & -0.0419 & 0.0051 \end{bmatrix}$$

It can be seen from J that a solution (U_1,V_6), (U_2,V_1), (U_3,V_9), (U_4,V_4), (U_5,V_8), (U_6,V_5), (U_7,V_3), (U_8,V_7) has the highest comprehensive benefit, whose value is 1.8251 (the sum of dark elements in matrix J), and whose corresponding objective value is (129.66,14.39,58.08).The results with MOPSO are shown in Table 2. Compared with the result of reference [4], it can be concluded that the method proposed in this paper can produce a set of optimized solutions which isn't dominated by the method proposed in reference [4] and which have a good distribution, hence meeting different decision-makers' preferences. On the other hand, supposing one knows the decision-maker's preference exactly, the method proposed in reference [4] produces only a satisfactory solution in each run. Therefore much operation is needed to meet different decision-makers' preferences, and hence the computational cost of the algorithm increases.

Table 2 Results with MOPSO on an eight-project ten-site location problem

	Solutions of location problem	Objectives value
1	$(U_1,V_7),(U_2,V_5),(U_3,V_2),(U_4,V_4),(U_5,V_8),(U_6,V_7 6,(U_7,V_3),(U_8,V_{10})$	(76.34, 21.1, 44.4)
2	$(U_1,V_7),(U_2,V_4),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_5),(U_7,V_3),(U_8,V_6)$	(147.57, 12.69, 56.52)
3	$(U_1,V_7),(U_2,V_4),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_6),(U_7,V_3),(U_8,V_1)$	(124.65, 17.22, 29.54)
4	$(U_1,V_6),(U_2,V_4),(U_3,V_2),(U_4,V_{10}),(U_5,V_8),(U_6,V_7),(U_7,V_3),(U_8,V_1)$	(60.14, 12.82, 68.64)
5	$(U_1,V_6),(U_2,V_4),(U_3,V_3),(U_4,V_9),(U_5,V_8),(U_6,V_5),(U_7,V_{10}),(U_8,V_7)$	(136.12, 15.49, 47.43)
6	$(U_1,V_7),(U_2,V_4),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_5),(U_7,V_3),(U_8,V_1)$	(109.64, 18.66, 40.82)
7	$(U_1,V_6),(U_2,V_5),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_1)$	(55.35, 16.71, 61.34)
8	$(U_1,V_1),(U_2,V_2),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_{10})$	(129.99, 10.62, 57.84)
9	$(U_1,V_6),(U_2,V_2),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_1),(U_7,V_5),(U_8,V_{10})$	(95.1, 19.48, 44.35)
10	$(U_1,V_7),(U_2,V_5),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_6),(U_7,V_3),(U_8,V_1)$	(82.41, 20.35, 35.53)

5 Conclusion

Multi-objective multi-site location problems are complicated optimization problems. Because particle swarm optimization has such good performance characteristics as easy implantation, simple parameter settings and fast convergence, multi-objective particle swarm optimization based on the minimal particle angle is used in this paper to solve the problems successfully by equivalent probability matrix. It can be seen by optimizing an eight-project -ten-site location problem that the method proposed in this paper can produce a set of satisfactory solutions with low computational cost, giving decision-makers many chances to choose.

References

[1] Zhou Q X, Zhang D X. Fuzzy constrained relationship intersection chart optimized decision-making model with multi-objects and its application. *System Engineering & Electronic Technology*, 1997 (11): 4–8. (In Chinese)

[2] Chen S Y, Xiong D Q, Zhao Y Q. Fuzzy relationship optimized decision-making theory for constrained multi-objective systems and its application in site selection of port works. *Systems Engineering—Theory & Practice*, 1995, 15 (2): 41–48. (In Chinese)

[3] Song Y X, Chen M Y, Zhang S H. Multi-objective decision-making method for site selecting of projects based on fuzzy assignment. *Systems Engineering—Theory & Practice*, 2002, 22 (1): 108–111. (In Chinese)

[4] Dai W Z. Multi-objective decision-making for site selecting of projects and its application. *Systems Engineering—Theory & Practice*, 2002, 24 (9): 79–84. (In Chinese)

[5] Kennedy J, Eberhart R C. Particle swarm optimization. *In: Proceedings of IEEE International Conference on Neural Networks*, 1995: 1942–1948.

[6] Pang W, Wang K P, Zhou G C et al. Fuzzy discrete particle optimization for solving traveling salesman problem. *In: Proceeding of the International Conference on Computer and Information Technology*, Wuhan, China, IEEE Computer Society, 04: 796–800.

[7] Coello C C A, Lechuga M S. MOPSO: A proposal for multiple objective particle swarm optimization. *In: Proceedings of the 2002 IEEE Congress on Evolutionary Computation*, Honolulu, 2002: 1051–1056.

[8] Fieldsend J E, Singh S. A multi-objective algorithm based upon particle swarm optimization. *In: Proceedings of the U K Workshop on Computational Intelligence*, 2002: 34–44.

[9] Mostaghim S, Teich J. Strategies for finding good local guides in multi-objective particle swarm optimization (MOPSO). In: *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, Indianapolis, 2003: 26–33.

[10] LI X D. A non-dominated sorting particle swarm optimizer for multi-objective optimization. *In: Proceedings of Genetic and Evolutionary Computation (GECCO)*, 2003: 37–48.

[11] Gong D W, Zhang Y. Multi-objective particle swarm optimization based on minimal particle angle. *Lecture Notes in Computer Science*, 2005, 3644 (1): 571–580.